Amendments to the Claims:

No claims are amended. All pending claims are reproduced below.

In the Claims:

1. (Original) A method for computing a diversity measure for a predetermined combinatorial structure C having n elements, the method comprising steps of:

(a) identifying M substructures c_1 through c_M each having m elements from among the n elements of the predetermined combinatorial structure C, where M equals n! / [(n-m)! m!];

(b) for each substructure c_i, for i from 1 to M, determining a number n_i of the M substructures c_1 through c_M that are similar to the substructure c_i; and

(c) computing a first entropy $\Phi(m)$ based upon all the numbers n_i computed during step (b) and based upon M in computed step (a);

2. (Original) A method as in claim 1, further comprising the steps of:

(d) repeating steps (a) and (b) with m+1 substituted for m;

(e) computing a second entropy $\Phi(m+1)$ based upon all the numbers n_i and M computed during step (d); and

(f) subtracting the second entropy $\Phi(m+1)$ from the first entropy $\Phi(m)$ to produce the diversity measure.

3. (Original) A method as in claim 2, wherein steps (c) and (e) comprise the steps of:

for each i from 1 to M:

computing a fraction F_i by dividing n_i by M; and

computing a logarithm of fraction F_i;

computing a sum by adding all logarithms of fractions F_i for i from 1 to M; and

dividing the sum by M.

4. (Original) A method as in claim 2, wherein step (b) comprises the steps of, for each substructure c_i for i from 1 to M:

for each substructure c_j for j from 1 to M:

computing a distance function $d(c_i,c_j)$ representing a measure of a difference between substructure c_i and substructure c_j;

comparing the distance function $d(c_i,c_j)$ to a threshold; and

determining the substructures c_i and c_j to be similar if and only if the distance function $d(c_i,c_j)$ is less than the threshold.

5. (Original) A method as in claim 2, wherein steps (c) and (e) comprise the steps of:

for each distinct substructure c_i:

computing a frequency f_i by dividing n_i by M;

computing a logarithm of frequency f_i; and

computing a product by multiplying the frequency f_i and the logarithm of frequency f_i; and

computing a sum by adding all products of the frequencies f_i and the logarithms of frequencies f_i.

6. (Original) A method as in claim 2, wherein step (b) comprises the steps of:

for each substructure c_i for i from 1 to M:

monotonically renumbering m elements of c_i from 1 to m; and

for each substructure c_j for j from 1 to M:

monotonically renumbering m elements of c_j from 1 to m; and

determining the substructures c_i and c_j to be similar if and only if they are identical.

7. (Original) A method as in claim 2, wherein step (b) comprises the steps of:

for each substructure c_i for i from 1 to M:

monotonically renumbering m elements of c_i from 1 to m; and

for each substructure c_j for j from 1 to M:

monotonically renumbering m elements of c_j from 1 to m; and

determining the substructures c_i and c_j to be similar if and only if they are identical or isomorphic.

8. (Original) A method as in claim 2, wherein steps (c) and (e) comprise the steps of:

for each distinct substructure c_i:

computing a frequency f_i by dividing n_i by M;

computing a quotient by dividing the frequency f_i by an expected frequency p_i;

computing a logarithm of quotient q_i; and

computing a product by multiplying the frequency f_i and the logarithm of quotient q_i; and

computing a sum by adding all products of the frequencies f_i and the logarithms of quotients q_i.

9. (Original) A method as in claim 2, wherein the predetermined combinational structure C comprises a linked graph, wherein the n elements comprise n nodes.

10. (Original) A computer readable storage medium, comprising:

computer readable program code embodied on said computer readable storage medium, said computer readable program code for programming a computer to perform a method for computing a diversity measure for a predetermined combinatorial structure C having n elements, the method comprising steps of:

(a) identifying M substructures c_1 through c_M each having m elements from among the n elements of the predetermined combinatorial structure C, where M equals n! / [(n-m)! m!];

(b) for each substructure c_i, for i from 1 to M, determining a number n_i of the M substructures c_1 through c_M that are similar to the substructure c_i; and

(c) computing a first entropy $\Phi(m)$ based upon all the numbers n_i computed during step (b) and based upon M in computed step (a);

11. (Original) A computer readable storage medium as in claim 10, the method further comprising the steps of:

(d) repeating steps (a) and (b) with m+1 substituted for m;

(e) computing a second entropy $\Phi(m+1)$ based upon all the numbers n_i and M computed during step (d); and

(f) subtracting the second entropy $\Phi(m+1)$ from the first entropy $\Phi(m)$ to produce the diversity measure.

12. (Original) A computer readable storage medium as in claim 11, wherein steps (c) and (e) comprise the steps of:

for each i from 1 to M:

computing a fraction F_i by dividing n_i by M; and

computing a logarithm of fraction F_i;

computing a sum by adding all logarithms of fractions F_i for i from 1 to M; and

dividing the sum by M.

13. (Original) A computer readable storage medium as in claim 11, wherein step (b) comprises the steps of, for each substructure c_i for i from 1 to M:

for each substructure c_j for j from 1 to M:

computing a distance function $d(c_i,c_j)$ representing a measure of a difference between substructure c_i and substructure c_j;

comparing the distance function $d(c_i,c_j)$ to a threshold; and

determining the substructures c_i and c_j to be similar if and only if the distance function $d(c_i,c_j)$ is less than the threshold.

14. (Original) A computer readable storage medium as in claim 11, wherein steps (c) and (e) comprise the steps of:

for each distinct substructure c_i:

computing a frequency f_i by dividing n_i by M;

computing a logarithm of frequency f_i; and

computing a product by multiplying the frequency f_i and the logarithm of frequency f_i; and

computing a sum by adding all products of the frequencies f_i and the logarithms of frequencies f_i.

15. (Original) A computer readable storage medium as in claim 11, wherein step (b) comprises the steps of:

for each substructure c_i for i from 1 to M:

monotonically renumbering m elements of c_i from 1 to m; and

for each substructure c_j for j from 1 to M:

monotonically renumbering m elements of c_j from 1 to m; and

determining the substructures c_i and c_j to be similar if and only if they are identical.

16. (Original) A computer readable storage medium as in claim 11, wherein step (b) comprises the steps of:

for each substructure c_i for i from 1 to M:

monotonically renumbering m elements of c_i from 1 to m; and

for each substructure c_j for j from 1 to M:

monotonically renumbering m elements of c_j from 1 to m; and

determining the substructures c_i and c_j to be similar if and only if they are identical or isomorphic.

17. (Original) A computer readable storage medium as in claim 11, wherein steps (c) and (e) comprise the steps of:

for each distinct substructure c_i:

computing a frequency f_i by dividing n_i by M;

computing a quotient by dividing the frequency f_i by an expected frequency p_i;

computing a logarithm of quotient q_i; and

computing a product by multiplying the frequency f_i and the logarithm of quotient q_i; and

computing a sum by adding all products of the frequencies f_i and the logarithms of quotients q_i.

18. (Original) A computer readable storage medium as in claim 11, wherein the predetermined combinational structure C comprises a linked graph, wherein the n elements comprise n nodes.

19. (Original) A computer system, comprising:

a processor; and

a processor readable storage medium coupled to the processor having processor readable program code embodied on said processor readable storage medium, said processor readable program code for programming the computer system to perform a method for computing a diversity measure for a predetermined combinatorial structure C having n elements, the method comprising steps of:

(a) identifying M substructures c_1 through c_M each having m elements from among the n elements of the predetermined combinatorial structure C, where M equals n! / [(n-m)! m!];

(b) for each substructure c_i, for i from 1 to M, determining a number n_i of the M substructures c_1 through c_M that are similar to the substructure c_i; and

(c) computing a first entropy $\Phi(m)$ based upon all the numbers n_i computed during step (b) and based upon M in computed step (a);

20. (Original) A computer system as in claim 19, the method further comprising the steps of:

(d) repeating steps (a) and (b) with m+1 substituted for m;

(e) computing a second entropy $\Phi(m+1)$ based upon all the numbers n_i and M computed during step (d); and

(f) subtracting the second entropy $\Phi(m+1)$ from the first entropy $\Phi(m)$ to produce the diversity measure.

21. (Original) A computer system as in claim 20, wherein steps (c) and (e) comprise the steps of:

for each i from 1 to M:

computing a fraction F_i by dividing n_i by M; and

computing a logarithm of fraction F_i;

computing a sum by adding all logarithms of fractions F_i for i from 1 to M; and

dividing the sum by M.

22. (Original) A computer system as in claim 20, wherein step (b) comprises the steps of, for each substructure c_i for i from 1 to M:

for each substructure c_j for j from 1 to M:

computing a distance function $d(c_i,c_j)$ representing a measure of a difference between substructure c_i and substructure c_j;

comparing the distance function $d(c_i,c_j)$ to a threshold; and

determining the substructures c_i and c_j to be similar if and only if the distance function $d(c_i,c_j)$ is less than the threshold.

23. (Original) A computer system as in claim 20, wherein steps (c) and (e) comprise the steps of:

for each distinct substructure c_i:

computing a frequency f_i by dividing n_i by M;

computing a logarithm of frequency f_i; and

computing a product by multiplying the frequency f_i and the logarithm of frequency f_i; and

computing a sum by adding all products of the frequencies f_i and the logarithms of frequencies f_i.

24. (Original) A computer system as in claim 20, wherein step (b) comprises the steps of:

for each substructure c_i for i from 1 to M:

monotonically renumbering m elements of c_i from 1 to m; and

for each substructure c_j for j from 1 to M:

monotonically renumbering m elements of c_j from 1 to m; and

determining the substructures c_i and c_j to be similar if and only if they are identical.

25. (Original) A computer system as in claim 20, wherein step (b) comprises the steps of:

for each substructure c_i for i from 1 to M:

monotonically renumbering m elements of c_i from 1 to m; and

for each substructure c_j for j from 1 to M:

monotonically renumbering m elements of c_j from 1 to m; and

determining the substructures c_i and c_j to be similar if and only if they are identical or isomorphic.

26. (Original) A computer system as in claim 20, wherein steps (c) and (e) comprise the steps of:

for each distinct substructure c_i:

computing a frequency f_i by dividing n_i by M;

computing a quotient by dividing the frequency f_i by an expected frequency p_i;

computing a logarithm of quotient q_i; and

computing a product by multiplying the frequency f_i and the logarithm of quotient q_i; and

computing a sum by adding all products of the frequencies f_i and the logarithms of quotients q_i.

27. (Original) A computer system as in claim 20, wherein the predetermined combinational structure C comprises a linked graph, wherein the n elements comprise n nodes.